

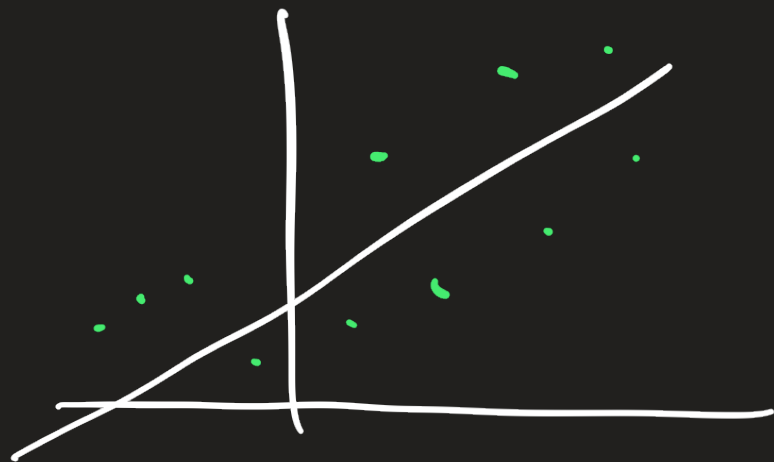
Lecture 3 ML & Optimization

- Parameterized ML Models
 - Gaussian
 - Bernoulli
 - Categorical
- Maximum Likelihood Estimation

- Gaussian model (for regression)

$$y|x \sim \mathcal{N}(\beta^T x, \sigma^2)$$

then $\mathbb{E}[y|x] = \beta^T x$



appropriate for regression when
observations are corrupted by
i.i.d. Gaussian noise and
 y depends linearly on x

Prediction: given x_{new} , take $\hat{y}_{\text{new}} = \beta^T x_{\text{new}}$

- Bernoulli noise model

$$y|x \sim \text{Bern}(p(x))$$

NB: we use to model ± 1 labels usually

useful for classification (e.g. spam detection)

$$\mathbb{E}[y|x] = p(x)$$

where we require $p(x) \in [0, 1]$. To ensure this, we take

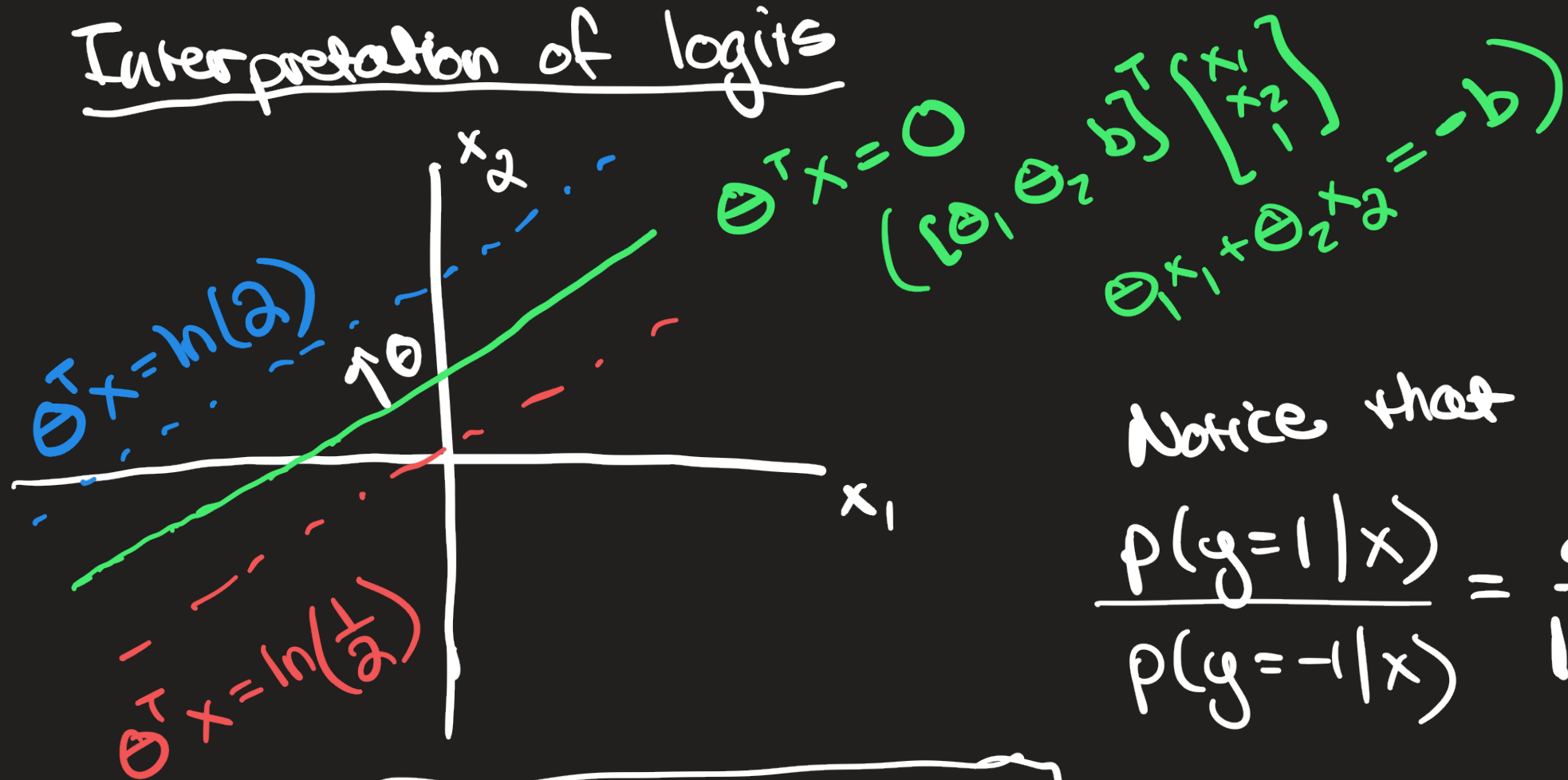
$$p(x) = \sigma(\underbrace{\theta^T x}_{\text{logit}})$$



$$\sigma(u) = \frac{1}{1 + e^{-u}}$$

Very used for binary classification (maybe w/ a nonlinear logit $f_{\theta}(x)$)

Interpretation of logits



Notice that

$$\frac{p(y=1|x)}{p(y=-1|x)} = \frac{\sigma(\theta^T x)}{1 - \sigma(\theta^T x)}$$

$$\log \frac{p(y=1|x)}{p(y=-1|x)} = \theta^T x$$

The log odds that $y=1$ increases as x moves in the direction θ

$$\begin{aligned} &= \frac{1}{1 + e^{-\theta^T x}} \cdot \frac{1}{1 - \left(\frac{1}{1 + e^{-\theta^T x}}\right)} \\ &= \frac{1}{1 + e^{-\theta^T x} - 1} \\ &= e^{\theta^T x} \end{aligned}$$

Prediction:

If $\sigma(\theta^T x) > \frac{1}{2}$ then predict $y = 1$

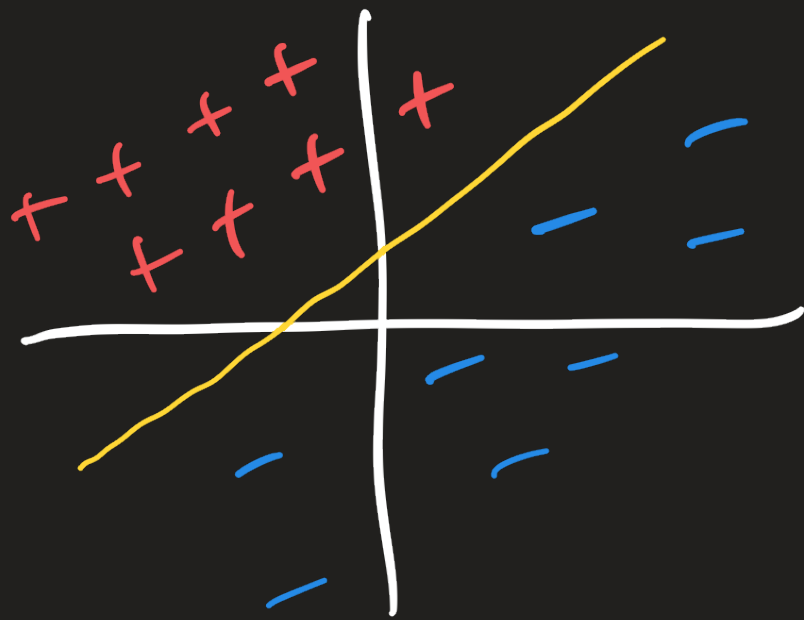
$\sigma(\theta^T x) < \frac{1}{2}$ then predict $y = -1$

Equiv:

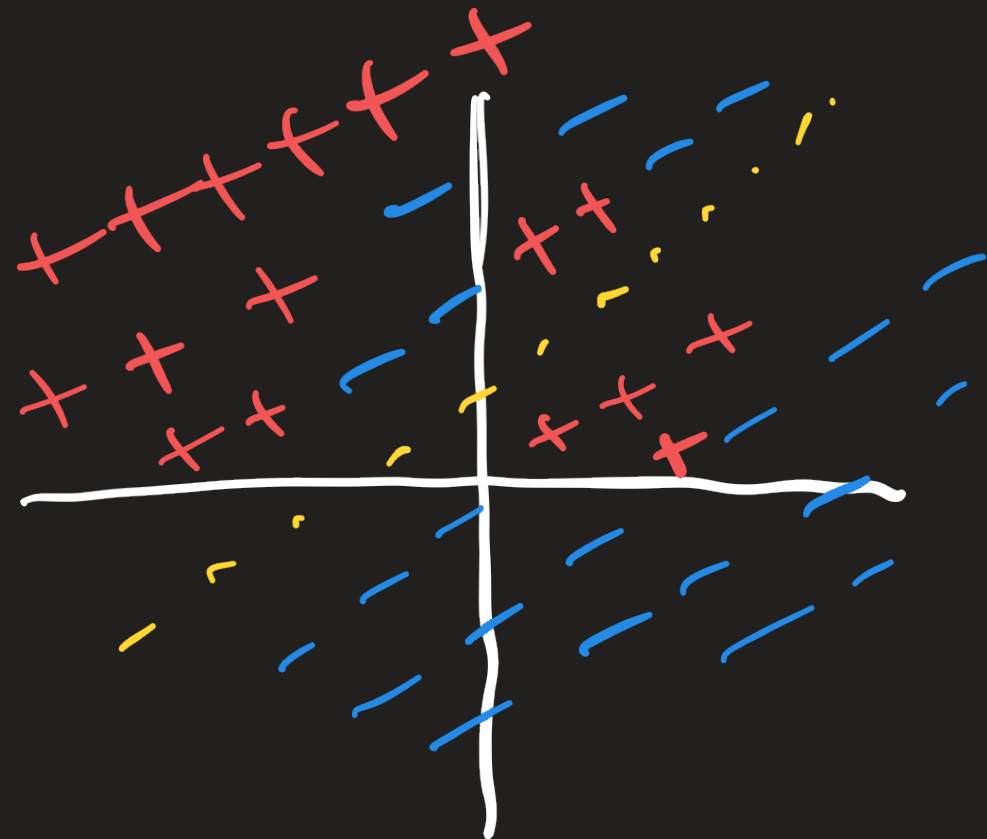
$\theta^T x > 0$ then predict $y = 1$

$\theta^T x < 0$ then predict $y = -1$

Difference b/w the SVM model & Bernoulli noise model for classification?



SVM: training data is always linearly separable



Bernoulli model:
training data is not required to be linearly separable

- Categorical regression (multiclass classification) model

A r.v. $z \sim \text{Categorical}(p_1, \dots, p_k)$ if

$$\mathbb{P}(z=i) = p_i \text{ for } i=1, \dots, k$$

where $\sum p_i = 1$ and $p_i \geq 0$

We use in ML for k-class classification

$$y|x \sim \text{Categorical}(p_1(x), \dots, p_k(x))$$

s.t. $\sum p_i(x) = 1$ and $p_i(x) \geq 0$

To ensure this take

$$p_i(x) = \frac{e^{\theta_i^T x}}{\sum_{j=1}^k e^{\theta_j^T x}} = \frac{e^{\theta_i^T x}}{Z_{\theta}(x)}$$

$$Z_{\theta}(x) = \sum_{i=1}^k e^{\theta_i^T x}$$

is called the partition function or the normalizing constant

logit of class i

Ex: Given pixels $x \in [256]^{m \times n}$

determine whether an image is in class $1, \dots, 1000$

Introduce some notation:

- Instead of letting $y = 1, \dots, k$ encode the classes as $y = e_1, \dots, e_k$ where $e_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \begin{matrix} \in \mathbb{R}^k \\ i^{\text{th}} \text{ position} \end{matrix}$

- Given a vector x , $e^x = \begin{bmatrix} e^{x_1} \\ \vdots \\ e^{x_k} \end{bmatrix}$

- Given $\theta_i \in \mathbb{R}^d$ for $i = 1, \dots, k$, define

$$\Theta = \begin{bmatrix} \theta_1^T \\ \vdots \\ \theta_k^T \end{bmatrix} \in \mathbb{R}^{k \times d}$$

- Let $\mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^k$ denote the vector of all ones

Then we have

$$p_{\theta}(x) = \begin{bmatrix} p_1(x) \\ \vdots \\ p_k(x) \end{bmatrix} = \begin{bmatrix} e^{\theta_1^T x} \\ \vdots \\ e^{\theta_k^T x} \end{bmatrix} \cdot \frac{1}{z_{\theta}(x)} = \frac{e^{\theta x}}{1^T e^{\theta x}}$$

$$= \text{softmax}(\theta x) \quad \text{where } \text{softmax}(v) = \frac{e^v}{1^T e^v}$$

$$\begin{array}{c} \overline{\theta x} \\ \left[\begin{array}{c} 30 \\ 1 \\ 0 \end{array} \right] \end{array} \xrightarrow{\text{softmax}} \begin{bmatrix} 1 \\ 2.54 \times 10^{-13} \\ 9.36 \times 10^{-14} \end{bmatrix}$$

$$\begin{bmatrix} 0.9 \\ 0.05 \\ 0.05 \end{bmatrix} \xrightarrow{\text{softmax}} \begin{bmatrix} 0.539 \\ 0.23 \\ 0.23 \end{bmatrix}$$

vector of
logits

$$\theta x = \begin{bmatrix} \theta_1^T x \\ \vdots \\ \theta_k^T x \end{bmatrix}$$

Notice:

$$\begin{aligned} E[y|x] &= e_1 p_1(x) + e_2 p_2(x) + \dots + e_k p_k(x) \\ &= \begin{bmatrix} p_1(x) \\ \vdots \\ p_k(x) \end{bmatrix} \end{aligned}$$

How to interpret the logits Θx ?

$$\frac{p(y=e_i|x)}{p(y=e_k|x)} = \frac{\text{softmax}(\Theta x)_i}{\text{softmax}(\Theta x)_k} = \frac{e^{\Theta_i^T x} / z_{\Theta}(x)}{e^{\Theta_k^T x} / z_{\Theta}(x)} \\ = e^{(\Theta_i - \Theta_k)^T x}$$

so $(\Theta_i - \Theta_k)^T x$ is the log odds of class i vs class k

How to predict y given x ?

$$\hat{y} = \underset{i}{\operatorname{argmax}} p(y=e_i|x) \\ = \underset{i}{\operatorname{argmax}} \Theta_i^T x$$

Assign the class w/ the largest logit

Maximum Likelihood Estimation

We have seen parametrized ML models for regression, binary classification, and multiclass classification corresponding to Gaussian, Bernoulli, and Categorical conditional distributions $y|x$

Most used formulations for regression & classification in ML, even for nonlinear models

Q: how to fit these models, given data?

A: Maximum Likelihood Estimation (MLE)

MLE i.i.d.!

Assume our n training data $\{(x_i, y_i)\}_{i=1}^n$ follows a parametrized distribution

$$y|x \sim p_{\theta}(y|x)$$

To estimate the optimal parameters θ , we use the maximum likelihood principle

"Choose θ so the joint density of the observations are maximized."

$$\begin{aligned}\hat{\theta} &= \underset{\theta}{\operatorname{argmax}} p_{\theta}(y_1, \dots, y_n | x_1, \dots, x_n) \\ &= \underset{\theta}{\operatorname{argmax}} p_{\theta}(y_1 | x_1) p_{\theta}(y_2 | x_2) \dots p_{\theta}(y_n | x_n) \\ &= \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^n p_{\theta}(y_i | x_i)\end{aligned}$$

likelihood of θ under observations

because data is i.i.d.

Numerically dealing w/ products is unwieldy, so use

$$\hat{\theta} = \operatorname{argmax}_{\theta} \prod_{i=1}^n p_{\theta}(y_i | x_i)$$

$$= \operatorname{argmax}_{\theta} \left(\prod_{i=1}^n p_{\theta}(y_i | x_i) \right)^{1/n}$$

$$= \operatorname{argmax}_{\theta} \frac{1}{n} \sum \log p_{\theta}(y_i | x_i)$$

$$= \operatorname{argmin}_{\theta} \underbrace{\frac{1}{n} \sum -\log p_{\theta}(y_i | x_i)}_{\text{called the negative log-likelihood of } \theta \text{ (NLL)}} := \mathcal{L}(\theta)$$

called the negative log-likelihood
of θ (NLL)

Issues in minimizing NLL

Q: does a minimizer exist (✓)

Q: is it unique (maybe)

Q: can we find it efficiently (✓)

- Ex: MLE for Gaussian model

$$y = \beta^T x + \varepsilon \quad \text{where } \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

or equiv

$$p_{\beta}(y_i | x_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \beta^T x_i)^2}{2\sigma^2}\right)$$

so

$$\mathcal{L}(\beta) = \frac{1}{n} \sum_{i=1}^n -\log p_{\beta}(y_i | x_i)$$

$$= \frac{1}{n} \sum_{i=1}^n -\left[\log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{(y_i - \beta^T x_i)^2}{2\sigma^2} \right]$$

$$= \log(\sqrt{2\pi}\sigma) + \frac{1}{n} \sum_{i=1}^n \frac{(y_i - \beta^T x_i)^2}{2\sigma^2}$$

Then

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \mathcal{L}(\beta)$$

$$= \underset{\beta}{\operatorname{argmin}} \log(\sqrt{2\pi}\sigma) + \frac{1}{n} \sum_{i=1}^n \frac{(y_i - x_i^T \beta)^2}{2\sigma^2}$$

$$= \underset{\beta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n (y_i - x_i^T \beta)^2$$

$$= \beta_{\text{OLS}}$$

ERM w/
 $\mathcal{L}(u, v) = (u - v)^2$

MLE of a Gaussian reg model's parameters
is OLS

- MLE for Bernoulli noise model

$$y|x \sim \text{Bern}(\sigma(\theta^T x))$$

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n -\log p_{\theta}(y_i|x_i)$$

Notice $p_{\theta}(1|x_i) = \sigma(\theta^T x_i)$

$$p_{\theta}(-1|x_i) = 1 - \sigma(\theta^T x_i) = \sigma(-\theta^T x_i)$$

because

$$1 - \sigma(u) = 1 - \frac{1}{1 + e^{-u}} = \frac{1 + e^{-u}}{1 + e^{-u}} - \frac{1}{1 + e^{-u}} = \frac{e^{-u}}{1 + e^{-u}}$$

$$= \frac{1}{e^u + 1} = \sigma(-u)$$



These imply

$$p_{\theta}(y_i|x_i) = \sigma(y_i \theta^T x_i)$$

$$\text{so } \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n -\log p_{\theta}(y_i | x_i)$$

$$= \frac{1}{n} \sum_{i=1}^n -\log \sigma(y_i \theta^T x_i)$$

$$= \frac{1}{n} \sum_{i=1}^n -\log \left[\frac{1}{1 + e^{-y_i \theta^T x_i}} \right]$$

$$= \frac{1}{n} \sum_{i=1}^n \log(1 + e^{-y_i \theta^T x_i})$$

ERM with $\ell(u, v) = \log(1 + e^{-uv})$

and

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \log(1 + e^{-y_i \theta^T x_i})$$

← this is
logistic
regression